IMPLEMENTING A GPS WAVEFORM UNDER THE SOFTWARE COMMUNICATION ARCHITECTURE

Alison Brown (NAVSYS Corporation, Colorado Springs, Colorado; abrown@navsys.com); Lynn Stricklan (NAVSYS Corporation, Colorado Springs, Colorado, lynns@navsys.com ; and David Babich (NAVSYS Corporation, Colorado Springs, Colorado, davidb@navsys.com)

ABSTRACT

SCA governs the structure and operation of software defined radios, enabling programmable radios to load waveforms, run applications, and network into an integrated system. Adherence to standards detailed in the SCA definition document allows hardware and software designers to know what equipment and programs to design. The SCA Hardware (HW) Framework tells the designer what minimum design specifications must be met by hardware devices. These specifications assure software written to the SCA guidance will run on SCA compliant hardware. Similar software specifications are provided for software applications. The core framework provides an abstraction layer between the waveform application and the software defined radio, enabling application porting to multiple vendors SDR products.

NAVSYS is engaged in creating an SCA compliant prototype for an embedded Global Positioning System (GPS) waveform in a software defined radio. The intent is to optimize GPS services by providing position and time as an embedded waveform within a Software Defined Radio rather than requiring additional GPS chip sets. This paper will cover the design of the GPS devices and prototype software defined radio (PowerPC processor and Xilinx FPGAs) used to implement and test the GPS waveform under the SCA. This application necessitates the ability to switch tasking and adjust to various mission types within the radio framework. Test results are included showing the ability to run the GPS waveform under the SCA and demonstrating the GPS waveform with performance in tracking the GPS satellites.

A discussion is also included on how the waveform could be ported to different SDRs running the SCA with different host processors and FPGAs. The flexibility of the design will allow SDR-enabled devices to be programmed to include GPS functionality running within the same radio that is supporting communications functions. An overview of some of the candidate Software Defined Radios that we can port the GPS waveform into is provided with a discussion on potential commercial applications within COTS SDRs. The open systems architecture provided by the SCA enables the addition of enhanced capabilities as technology advances through software upgrades. These could include the addition of new GPS codes and other GNSS services as these become available.

1. INTRODUCTION

The Software Communications Architecture (SCA) is an open architecture framework that tells designers how elements of hardware and software are to operate in harmony within a software defined radio. SCA is a standard adopted by the Object Management Group (OMG) and is being used for the Joint Tactical Radio System (JTRS) and is being promoted by the Software Defined Radio (SDR) Forum. SCA governs the structure and operation of software defined radios, enabling programmable radios to load waveforms, run applications, and be networked into an integrated system.

Through adherence to standards detailed in the SCA definition document, both hardware and software designers know what equipment and programs to design. The SCA Hardware (HW) Framework tells the designer what minimum design specifications must be met by hardware devices. These specifications assure software written to the SCA guidance will run on SCA compliant hardware. Similar software specifications are provided for software applications. The core framework provides an abstraction layer between the waveform application and the software defined radio, enabling application porting to multiple vendors SDR products.

NAVSYS is engaged in creating an SCA compliant prototype for an embedded Global Positioning System (GPS) waveform in a software defined radio. The intent is to optimize GPS services by providing position and time through a single piece of equipment and demonstrate the use of a precision GPS architecture within a network centric environment. This application necessitates the ability to switch tasking and adjust to various mission types within the radio's framework.

This paper describes the architecture of the prototype software defined radio used to test the GPS Waveform. The paper also covers the design of the GPS devices used to implement the GPS waveform under the SCA. Test

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **NOV 2006**	2. REPORT TYPE	3. DATES COVERED **00-11-2006 to 00-11-2006**
4. TITLE AND SUBTITLE **Implementing a GPS Waveform Under the Software Communication Architecture**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **NAVSYS Corporation,14960 Woodcarver Road,Colorado Springs,CO,80921**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**12**	

results are included showing the ability to run the GPS waveform under the SCA and demonstrating the GPS waveform's performance in tracking the GPS satellites.

2. JOINT TACTICAL RADIO SYSTEM (JTRS)

The initial target software defined radios for the GPS Waveform are the JTRS radios. The Joint Tactical Radio System (JTRS) includes a family of radios, waveforms and crypto algorithms designed under the software communications architecture (Figure 1). The JTRS radios are reprogrammable to run the family of waveforms shown in Figure 2 which extend from 2 MHz to 2 GHz. While the JTRS radios are capable of operating at the GPS frequencies (1.5 and 1.2 GHz), and GPS functionality is a requirement for providing time and position to these radios, GPS is currently planned as an embedded SAASM module rather than as a waveform in the JTRS architecture.

This paper describes a prototype GPS waveform that is being developed for potential application in the family of JTRS radios shown in Figure 3. This is being used to evaluate the radio resources needed to implement a GPS waveform and as a proof of concept to demonstrate the future benefits provided by deployment of a GPS waveform to support the JTRS timing and positioning requirements.

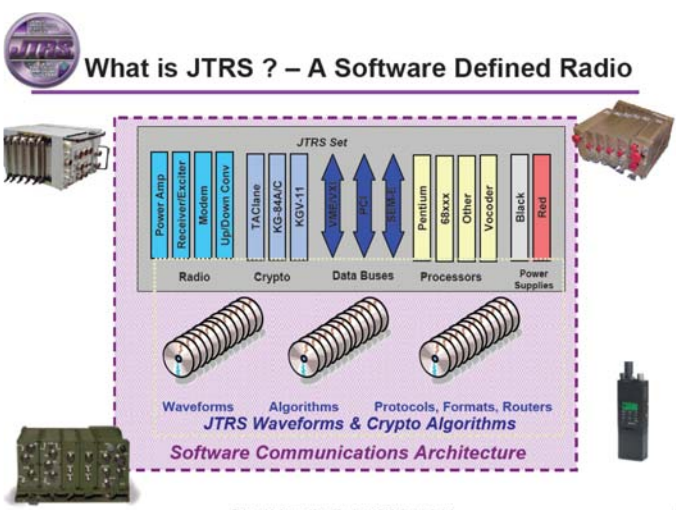

Figure 1 Joint Tactical Radio System (JTRS)[1]

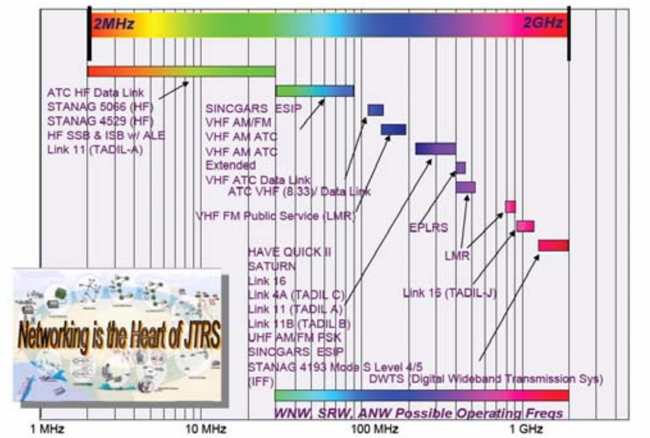

Figure 2 JTRS Waveforms

3. SOFTWARE COMMUNICATIONS ARCHITECTURE

An inherent aspect of Software Defined Radios (SDR) is the ability to redefine radio functionality by interchanging waveform software on one set of radio hardware. This allows for communications interoperability of the various military, domestic, and coalition forces. The industry demand for a clearly defined hardware, software, and network architecture to provide a standardized environment for the deployment of interoperable waveforms has been the driving force for standardization of an open architecture. This demand brought about the Software Communications Architecture (SCA).

Figure 3 Some JTRS Radios

The SCA was established by the JTRS JPO to provide a common open architecture for the development and deployment of software on software defined radio systems. SCA defines a common operating environment (OE) consisting of a Core Framework (CF), CORBA middleware, and a POSIX-based operating system (OS). The establishment of a common OE allows for the

deployment of waveform software across multiple radios, platforms and domains with little to no changes to the software. This reduces development time and cost, and improves radio network interoperability.

A more in depth view of the OE from the bottom up reveals how SCA accomplishes standardized interoperability. The OS provides the required underlying services and interfaces of the SCA compliant system. These requirements are defined by the SCA Application Environment Profile (AEP) [2]. The AEP is composed of a required set of C and POSIX standards of which a subset of each standard must be present. There are also optional components to the AEP which may or may not be present. In an SCA compliant system, the application developer is guaranteed to be provided with at least the minimum required AEP. The CF provides a standardized API that is an abstraction of the underlying hardware and software in the system [3]. The CF Interface Description Language (IDL) defines the standard set of access points to the OS for the application developer. Non-CORBA components can gain access to the OS via CORBA wrappers referred to as Adapters [7]. The CORBA middleware provides the logical software bus and the network service layer. It also provides a means for software distribution whereby processing components can be distributed across various resources on a network. The functional interaction of the three OE components alleviates the radio application developer from the responsibility of interoperability.

Configuration of the SDR is achieved by a set of XML descriptor profiles that describe the system domain. These XML files are referred to as the Domain Profile and consist of the following XML descriptor files; Software Package (SPD), Device Package Descriptor (DPD), Properties, Software Component Descriptor (SCD), Software Assembly Descriptor (SAD), Device Configuration Descriptor (DCD), Domain Manager Configuration and Profile [4].

Figure 4 through Figure 8 show some of the significant stages in deployment of a waveform in an SCA compliant system. Figure 4 is a depiction of the system start up procedure. The first SCA responsibility after start up is the creation of the Domain Manager. The Domain Manager is responsible for waveform creation and control. Once the Domain Manager is created, it registers itself with the CORBA naming service and event service, and creates its file system. After the Domain Manger registration, the Device Managers are created and registered with the Domain Manager. In an SCA compliant system, there can be any number of Device Managers.

Figure 5 shows the registration of the various devices (and services) within the system to the Device Managers. Each Device Manager controls a set of devices and services which are described by the Device Configuration Descriptor XML file (DCD) [4].

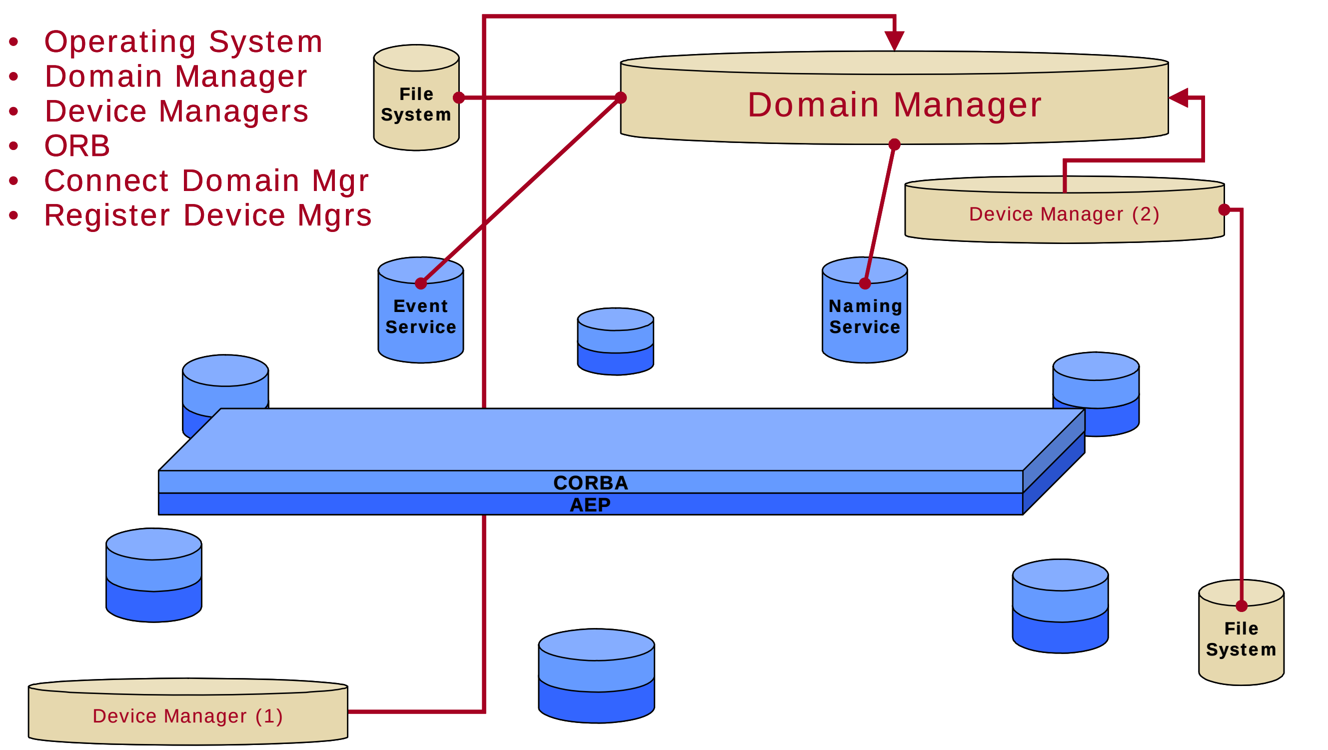

Figure 4 SCA System Start Up[5]

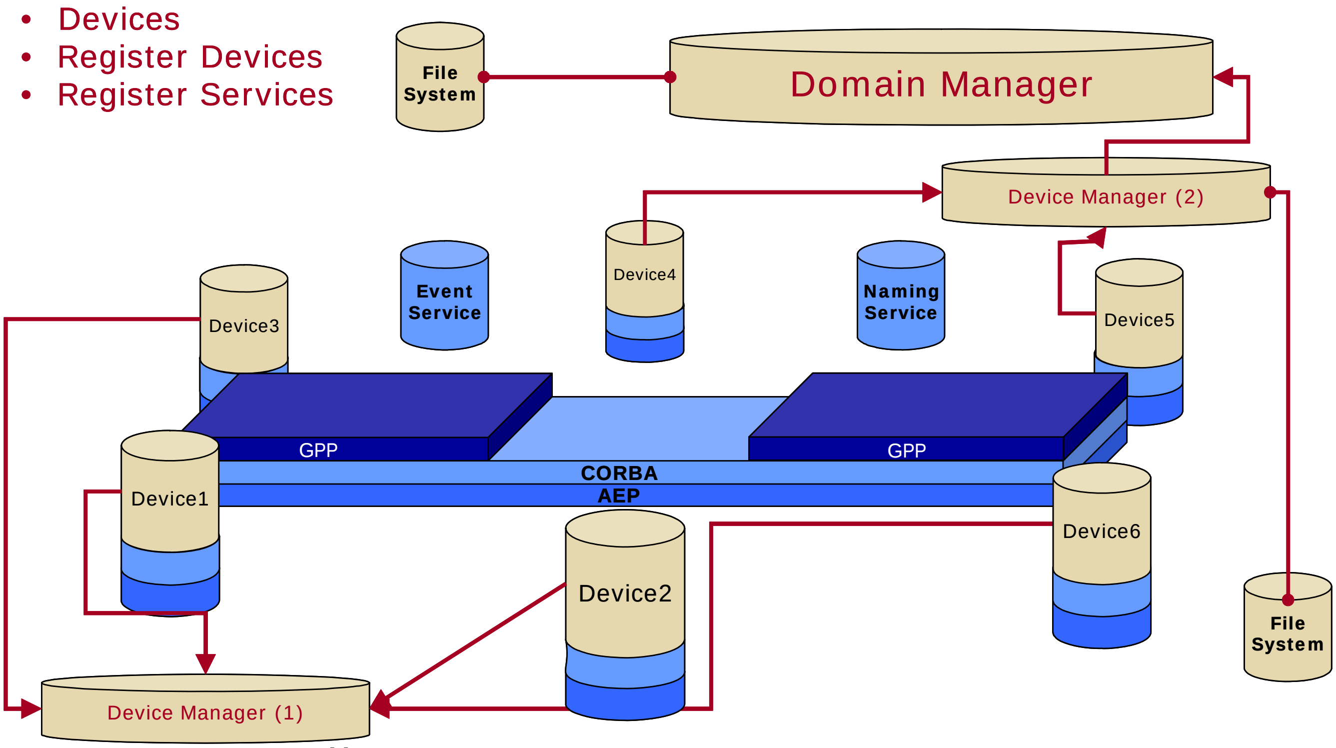

Figure 5 Device Registration[5]

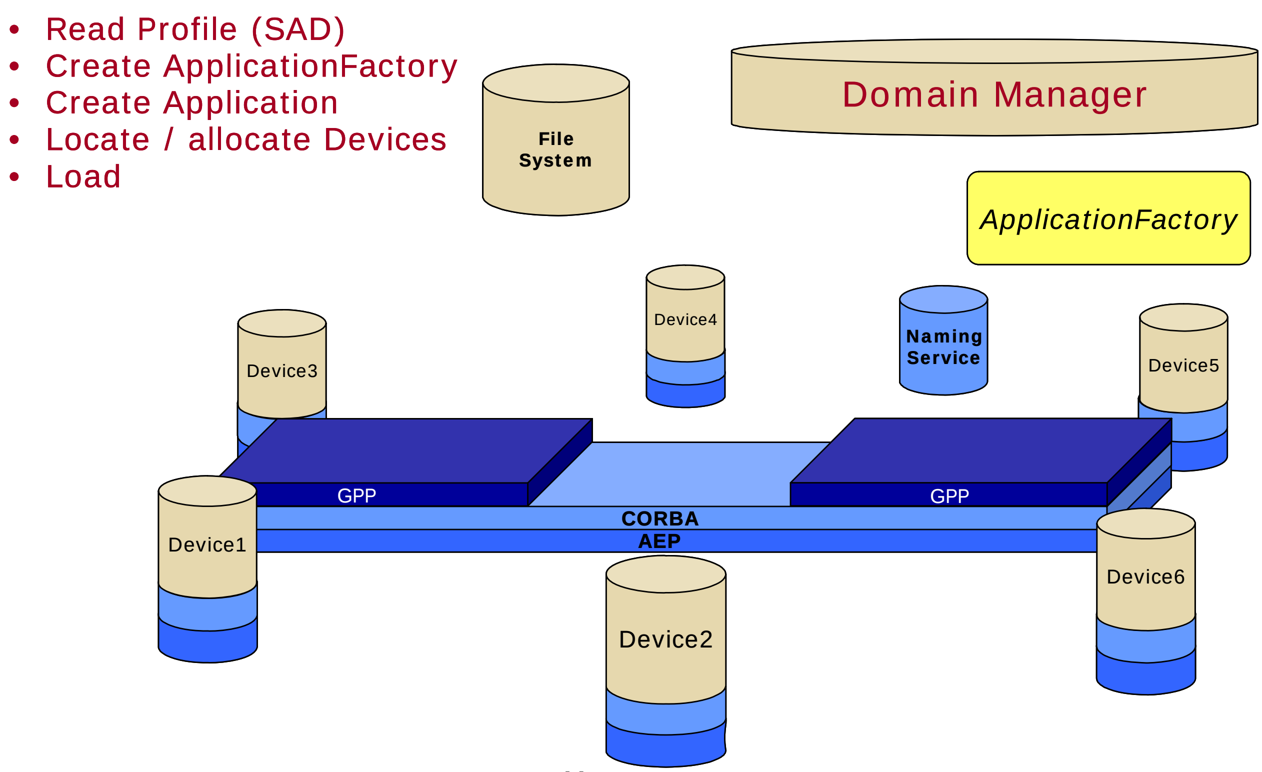

Figure 6 Waveform Instantiation and Deployment[5]

The next step in SCA waveform deployment is the creation of the waveform application and is depicted in Figure 6. The SDR administrative user requests the instantiation of the waveform application from the Application Factory via the Domain Manager. The Application Factory uses the Software Assembly Descriptor (SAD) XML profile to define which components comprise the waveform application. The waveform application is then created by the Application Factory, and is registered with the Domain Manager. During the creation of the waveform application, the application factory must locate and/or allocate the devices used by the application, and load the software components of the application to the appropriate processing element (GPP).

In the SCA Core Framework (CF) Device interface characteristics are defined using an inheritance scheme. Toward the top of the inheritance chain is the CF Resource interface. The Resource interface inherits the following CF interfaces; PortSupplier, LiveCycle, TestableObject, and PropertySet. Though a full explanation of the functionality encompassed within the various levels of the inheritance chain is more detail than is valuable for this paper, it is useful to note that Devices inherit the Resource interface. Thus by inheritance, all Devices are Resources. The Device interface adds allocate/de-allocate functionality, and adds device state information. Devices come in two flavors Loadable and Executable. Loadable Devices inherit the Device interface and add the ability to load and unload software components. Executable devices inherit the Loadable Devices interface and add execute and terminate functionality. Having defined the general composition of the Device interface, it is relevant to discuss the next stages of waveform deployment. Figure 7 shows the progression after the waveform loadable and executable devices are loaded and allocated. The Application Factory initializes the resources in the system and establishes connections between the relevant resources (devices) using functionality encompassed within the PortSupplier interface. The resources are then configured via the resource configure operation, and connections are established across the CORBA bus using the connectPort operation.

Figure 8 is a summary of the responsibilities of SCA in an SDR. SCA must instantiate and configure the waveform, establish connections between the relevant distributed resources, and when the user no longer has need of the waveform application, disconnect devices and tear down the application.

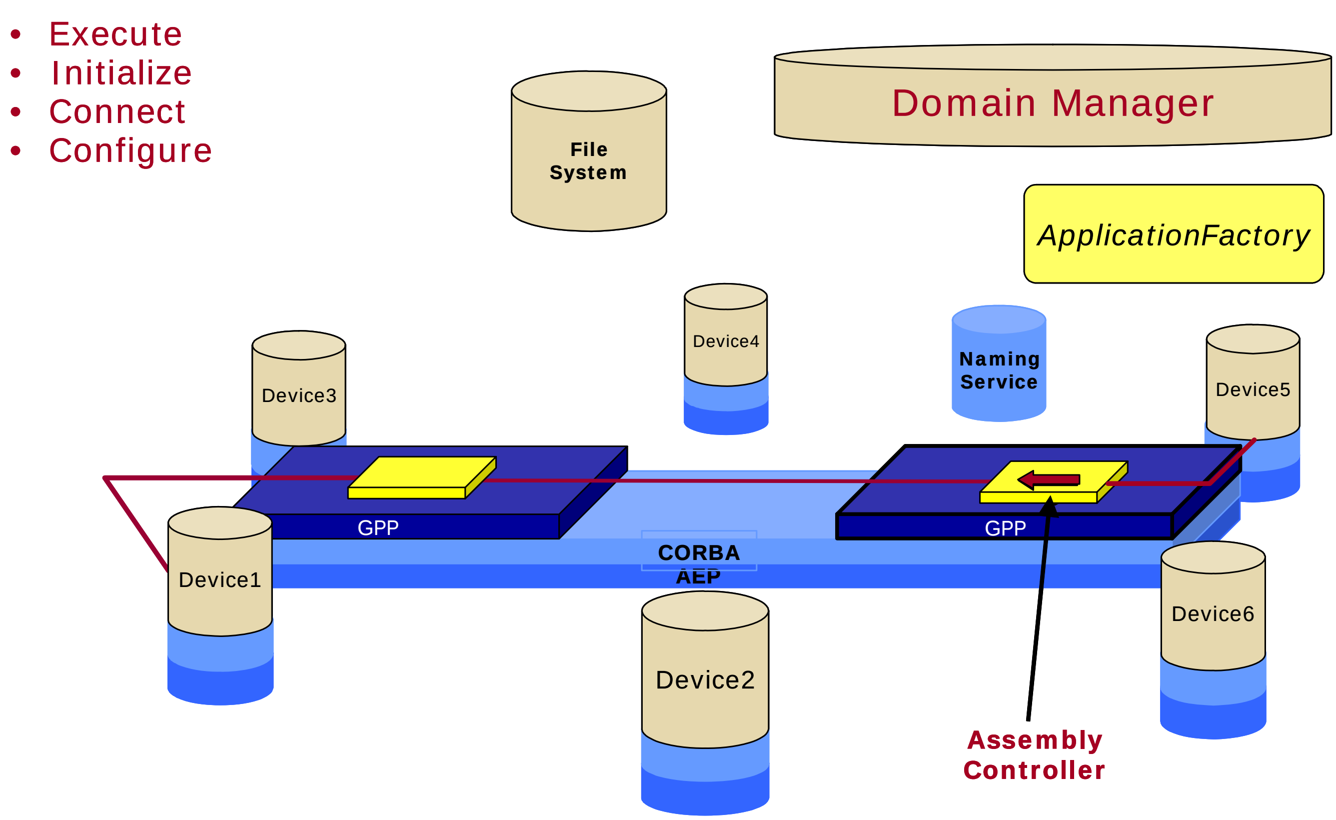

Figure 7 Waveform Instantiation and Configuration[5]

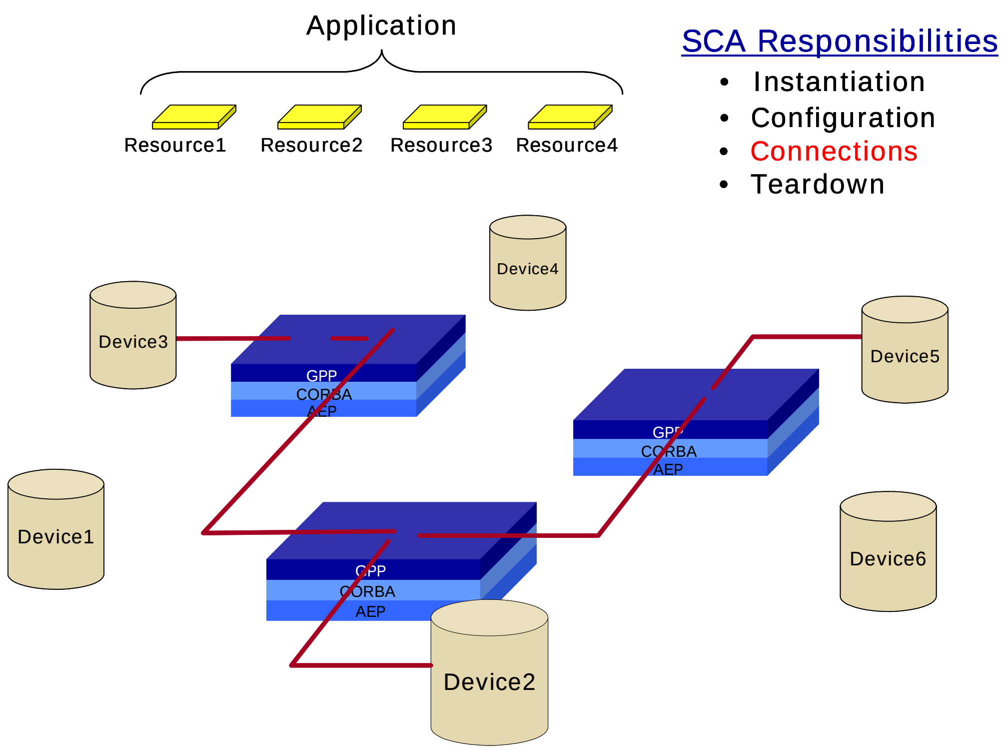

Figure 8 Summary of SCA Responsibilities[5]

4. SOFTWARE DEFINED RADIO TESTBED

Under a contract with NAVAIR NAVSYS has developed a GPS-Network-Assisted Positioning (GPS-NAP) testbed for the first generation deployment of a GPS waveform.

The GPS-NAP testbed includes the components illustrated in Figure 9 and is described in the proceeding paragraphs. The Testbed uses a PC/104 configuration as shown in Figure 10.

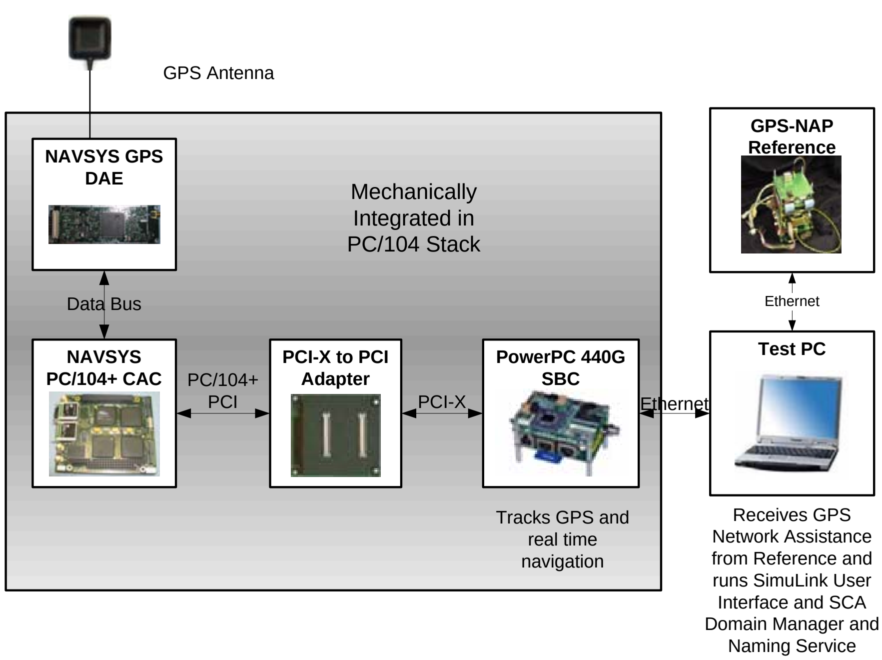

Figure 9 GPS-NAP Testbed Architecture [6]

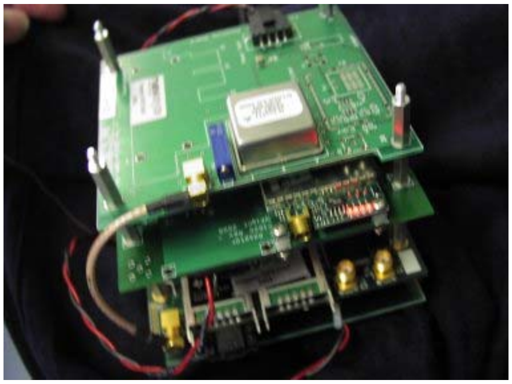

Figure 10 GPS-NAP Testbed

4.1. GPS-NAP Testbed Hardware

The design for the Phase II GPS-NAP Testbed is shown in Figure 11. This includes the sub-system components as illustrated in Figure 9 and described in the following sub-sections. These components were selected based on a representative JTRS radio design to allow the Testbed to be used as a development platform for a GPS-NAP waveform that could transition into the JTRS radios.

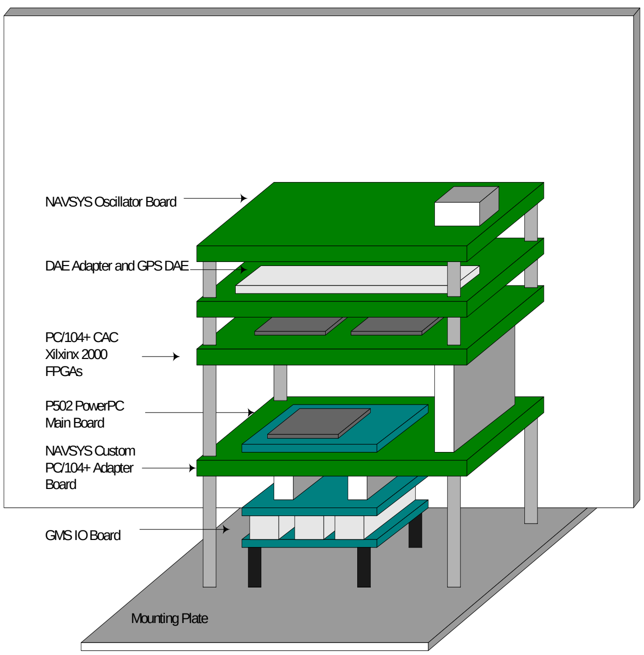

Figure 11GPS-NAP Software Defined Radio Components

4.2. PC/104 Mounting Board

The PC/104 Mounting Board is used to mount the reference oscillator, a 10 MHz Oven Controlled Crystal Oscillator (OCXO). The OCXO is a low phase-noise oscillator with temperature stability better than 0.1ppm. In addition to the oscillator, a multi-turn trim-pot is provided for absolute frequency adjustment. The clock stability and its performance under environmental conditions, such as temperature and vibration, are adequate to support the GPS tracking and performance requirements.

4.3. Digital Antenna Elements

The GPS Digital Antenna Element (DAE) seen in Figure 12 uses an active antenna which is powered by 5 VDC with current consumption of 50 mA maximum. The GPS DAE performs the functions of receiving and down converting the GPS L1 signals, digitizing these signals and outputting them to the PC/104 Correlator Accelerator Card (CAC) Board for processing. The GPS signals are input via an SMA-MCX connector from the active GPS antenna.

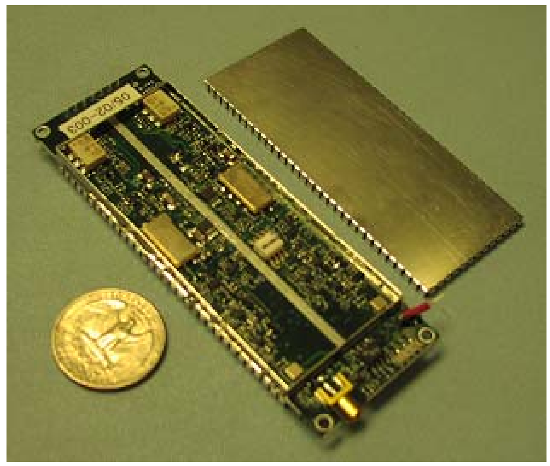

Figure 12 GPS L1/L2 DAE

4.4. PC/104 CAC Board

The PC/104 CAC Board shown in Figure 13 includes the FPGA which is used to perform the GPS code generation, code correlation and carrier mixing. The CAC firmware is downloaded from the PC/104 Single Board Computer (SBC).

Figure 13 PC/104 CAC FPGA Board

4.5. PowerPC Ultra Miniature Computer

The ultra miniature computer (UMC) is a General Micro Systems (GMS) P502 (or equivalent), which is a PowerPC 440Gx on an ultra small footprint (see Figure 14). The UMC is composed of a 667 MHz PowerPC processor with 256KB of L2 Cache on a footprint: 2.8" X 1.9" inches, two Gigabit Ethernet ports (502 Only), 256MB DDR SDRAM, 2 Serial ports, 16MB of Boot/user Flash, and 32KB of user Flash. The board is connected to the testbed with a custom PCI-X to PCI board in a PC/104 form factor. The operating temperature range is -40C to +85C.

Figure 14 Ultra Miniature Computer (UMC)

4.6. Power Supply

The UMC has low power requirements - 3.3VDC @ 300ma and 5-12VDC @ 0.5A and is delivered with a standard wall pluggable (“wall wart”) power converter.

The PC/104 Power Supply Board is an adjustable lab supplied DC power converter. This supplies the different voltages needed to drive the different PC/104 components.

The testbed units require 10.5 Watts average power at 12 VDC while in normal receive mode. Peak power during power-on is less than 19 Watts.

5. GPS WAVEFORM DESIGN

The generalized application components of the GPS waveform consist of a GPS receiver Digital Antenna Element (DAE) (or Modem RF front-end), GPS signal processing/correlation deployed on an FPGA or DSP (CAC), GPS (Receiver and Tracking) and Network-Assistance software application components deployed on a GPP, and in the case of P(Y) mode of operation the Precise Positioning Service (PPS) Security Module (SM) and Auxiliary Output Chip (AOC) functions performed within the JTRS crypto resources.

The GPS waveform is distributed amongst a number of hardware and software components in the system and is flexible in its utility of system resources. This makes it possible to deploy the waveform on varying SDR platforms with different resources available.

Figure 15 shows a data flow architectural diagram of SCA and the various functional components of a typical waveform application. Applying Figure 15 to the GPS waveform yields the DAE as the RF block utilizing a physical API to the GPS signal processing/correlation in an FPGA or DSP (in the testbed this is the CAC board). GPS Receiver, Tracking and Network-Assistance software application components are deployed on the Host GPP. The GPS applications running on the host GPP are coupled with the VHDL Modem Components through the CAC device driver.

The initial prototype waveform did not implement a full SCA architecture as we were concerned with basic functionality and evaluating resource loading. Utilizing the domain manager we included the ability to start and stop the radio service with the build and tear down processes associated with the waveform. We can also control the GPS waveform, providing functionality such as selecting different satellite channels, through the domain management user interface. The GPS-NAP software application components resides on the Power PC processor within the testbed. These are illustrated in Figure 16. Security Services for P(Y) processing are in development. The ACE/TAO software suite is used to provide CORBA services that are used by the Core Framework, either dmTK or SCARI.

The Core Framework is deployed with the Device Manager on the testbed. The Device Manager receives the networked requests from the Domain Manager proceeds to initiate the indicated action. The Naming Service and Domain Manager are currently deployed on the Test PC. This dual processor implementation emulates the division of resources between the dual red and black processors as shown in Figure 15. The logical software bus extends across two machines. CORBA makes network communications on the logical software bus transparent. It is also possible to combine these services on a single machine.

The GPS waveform initiates as a single aggregate component (Figure 17). The internal parts are divided into the modem, track and navigate. The CAC driver receives the GPS signal from the antenna. The CAC handles the carrier and code frequency mixing from the antenna data and then correlates these signals against C/A code generated. The driver is loaded when the radio is initiated and forwards data to real time track and non real time track. The real time track component optimizes range measurements. The non real time side refines time in conjunction with the solution message as a part of the ongoing position calculations. The non real time track contains the threads for track exec, receiver manager and

CORBA Comm for transferring range, and navigation data messages. Hybrid Navigator generates real time navigation solutions. The Test PC also runs two additional applications from Simulink a user interface and calculator for navigation and set of controls to start/stop the radio and select satellite channel.

The GPS waveform is configured at initiation using XML defined parameters within the Profile (.PRF) file. These include configuration parameters such as the IF frequency and sample rate of the RF-to-digital conversion, resource dependent parameters such as the number of satellite channels to be run, and optimization parameters such as Doppler search windows and the horizon limit to be used for satellite selection.

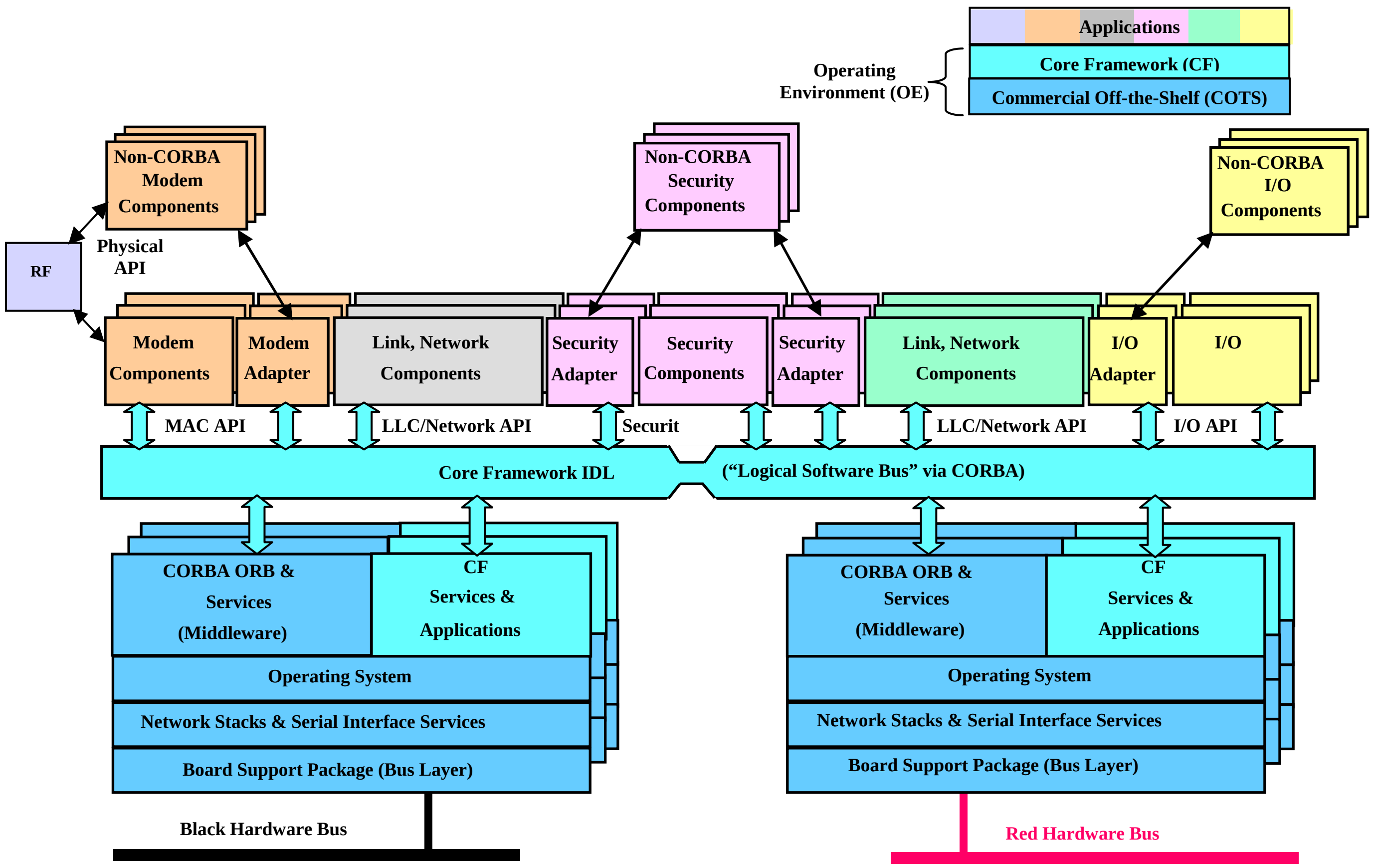

Figure 15: Software Communications Architecture and Operating Environment [7]

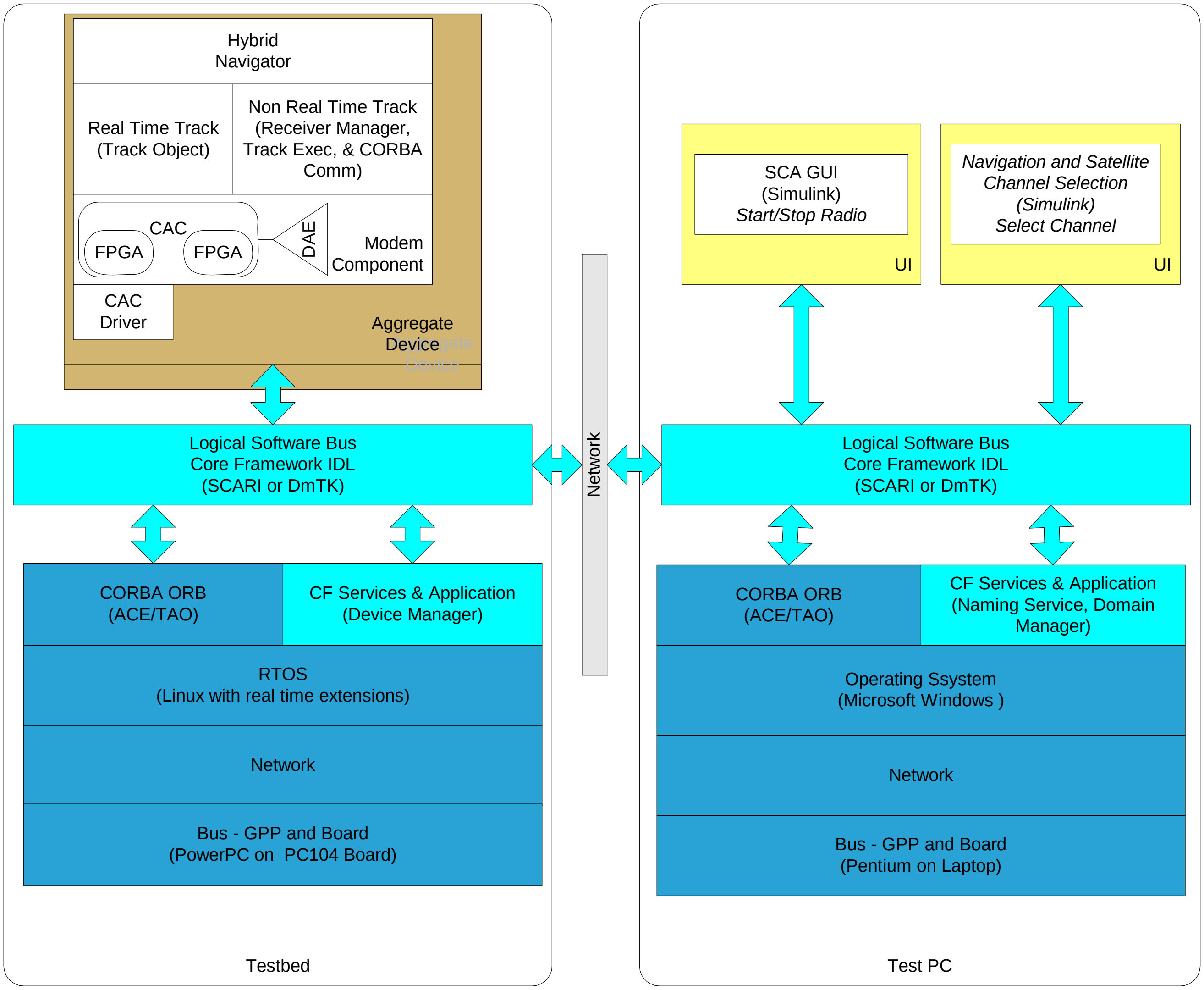

Figure 16 First SCA GPS Waveform Implementation

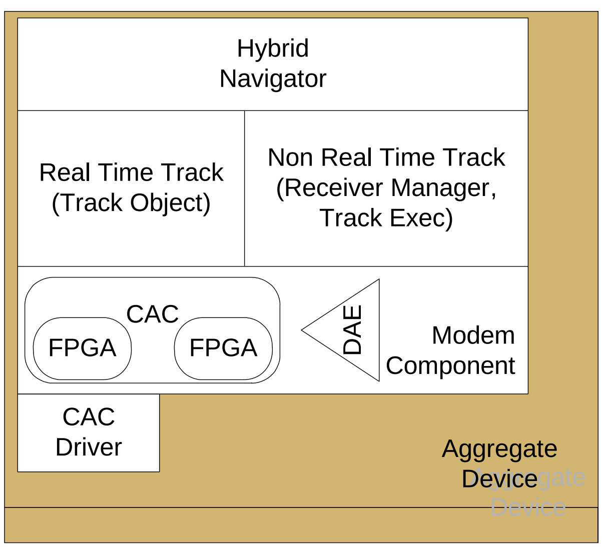

Figure 17 GPS Waveform Aggregate Device

6. SCA GPS WAVEFORM TESTING

We successfully tracked up to 4 satellites using the PC/104 CAC and PC/104 Power PC processor board. The tracking results are shown in the following figures. Figure 18 shows the change of the tracking state. In less than 20 seconds, all 4 satellites are in the highest state (State 630 indicates code and carrier lock and NAV data available).

Figure 19 shows the C/No of the satellites. PRN1, PRN 20, and PRN 23 all have reasonable C/No. The comparable low C/No of PRN24 is due to its low elevation angle.

Figure 20 shows the Pseudorange plus Carrier phase for each satellite, which indicates the multipath and noise level in the tracking channel. The PR+CPH are within 2 meters for PRN1, 20, and 23, which is reasonable for the noise + multipath in the tracking channel. PRN24 is low elevation satellite, which has strong multipath and causes PR+CPH to be higher than other channels.

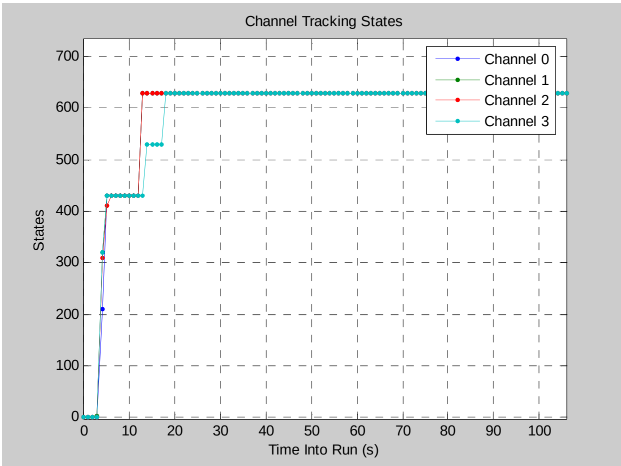

Figure 18 Change in Tracking States

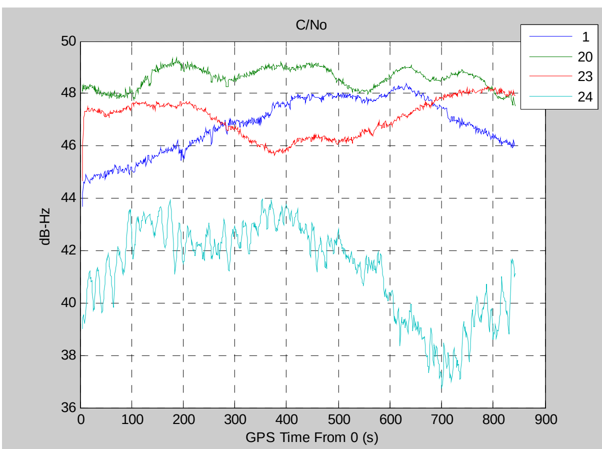

Figure 19 Satellite C/No

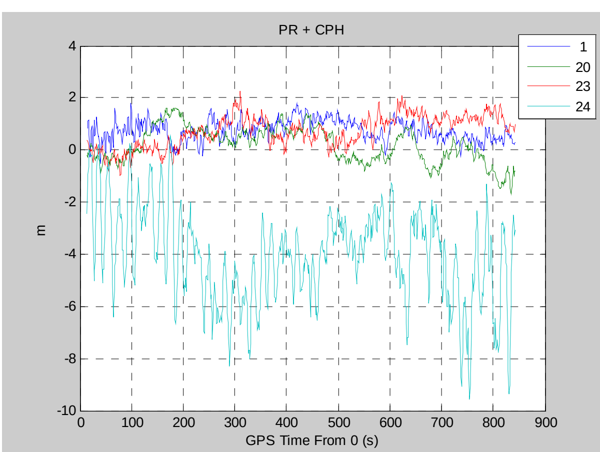

Figure 20 Pseudorange plus Carrier

6.1. CORBA COMMUNICATION TEST

The SCA architecture allows for communication to other User Interface functions through the CORBA interface Figure 21. To test this functionality we installed Simulink on the Reference PC and used this to display and process in real-time the CORBA messages provided by the GPS waveform. Figure 21 through Figure 22 summarize the results of this testing. This shows the real-time GUI implemented in Simulink to show the GPS-NAP tracking status. We were also able to process the raw track data to calculate navigation solutions using Simulink and Matlab. In Figure 22, we see a scatter diagram of our navigation solutions. Our horizontal and vertical accuracy are 1.5 meters and 2 meters (1-sigma) respectively.

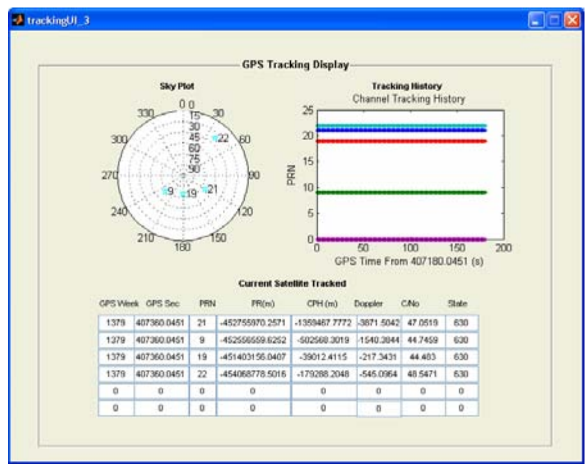

Figure 21 Simulink Tracking Display

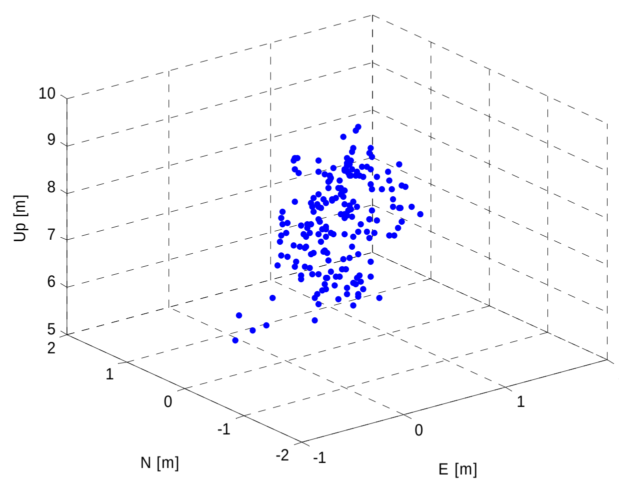

Figure 22 LLA Scatter Diagram

7. CONCLUSIONS

The prototype SCA GPS Waveform described in this paper has demonstrated that it is feasible to embed GPS as a waveform within software defined radios such as the Joint Tactical Radio System. The initial GPS waveform was capable of tracking four GPS satellites using a SDR with a PowerPC GPP and Spartan-3 FPGAs as resources. Further optimization in the design is expected to allow additional satellite channels to be handled. We are in the process of upgrading the C/A code waveform to add P(Y) code tracking using the SCA security resources. This prototype GPS waveform can be used as the basis for implementing a fully SCA compliant GPS waveform for use with JTRS radios or for commercial SDR applications.

Some of the benefits to running a GPS waveform under the software communications architecture are summarized below.

- The size, weight and power of the radio is reduced by avoiding the need for an embedded GPS device.
- CORBA interface simplifies GPS integration with other applications
- The development cost for vendors who wish to include GPS functionality in an SCA based radio product is reduced.
- Network assistance can be provided through the radio to improve the TTFF of the embedded GPS waveform and the accuracy of the GPS solution.
- Precise sub-nanosecond timing is available to other waveforms based on the GPS waveform calibration of the internal time reference.
- The GPS waveform can run as a background waveform, providing single channel radios the benefits of GPS without tearing down the current running waveform, provided that the system resources are available to run the GPS waveform.

8. ACKNOWLEDGMENTS

The authors would like to acknowledge the support of Naval Air Warfare Center, Patuxent River, Maryland, who has provided funding to support the development of this technology.

9. REFERENCES

[1] http://enterprise.spawar.navy.mil/UploadedFiles/MIDS_Overview.pdf

[2] Joint Tactical Radio Systems (JTRS) Joint Program Office, "JTRS-5000SCA Appendix B rev 3.0, Appendix B. SCA Application Environment Profile", August 27, 2004

[3] Raytheon, "Joint Tactical Radio Systems (JTRS) SCA Developer's Guide rev 1.1", June 18, 2002

[4] Joint Tactical Radio Systems (JTRS) Joint Program Office, "JTRS-5000SCA Appendix D rev 3.0, Appendix D. DOMAIN PROFILE", August 27, 2004

[5] "SCA Overview," Dr. Brian Salisbury, Manager, SCA JTRS Standards – JPEO JTRS, Unlimited Distribution October 3, 2005.

[6] A. Brown, C. Matthews, M. Loving, Y. Lu, D. Babich, T. Lehmpuhl, "GPS Network-Assisted Positioning C/A Code Tracking/Navigation Technical Demonstration Report," 2006, NAVSYS Document No. GPS-NAPII-06-059

[7] Joint Tactical Radio Systems (JTRS) Joint Program Office, "Software Communications Architecture Specification JTRS-5000 SCA V3.0," August 27, 2004